

BUNDESREPUBLIK DEUTSCHLAND

10. 03. 2004



EP 04/50165

REC'D 23 APR 2004

WIPO

PCT

**PRIORITY
DOCUMENT**
SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH RULE 17.1(a) OR (b)

**Prioritätsbescheinigung über die Einreichung
einer Patentanmeldung**

Aktenzeichen:

103 07 698.0

Anmeldetag:

21. Februar 2003

Anmelder/Inhaber:

Robert Bosch GmbH, 70442 Stuttgart/DE

Bezeichnung:Steuergerät und Computerprogramm zum Steuern
eines Antriebsaggregates eines Fahrzeugs**IPC:**

B 60 R, B 60 K

Die angehefteten Stücke sind eine richtige und genaue Wiedergabe der ursprünglichen Unterlagen dieser Patentanmeldung.

München, den 2. März 2004
Deutsches Patent- und Markenamt
Der Präsident
Im Auftrag

5 21.02.2003
Robert Bosch GmbH, 70442 Stuttgart

10 Steuergerät und Computerprogramm zum Steuern eines
Antriebsaggregates eines Fahrzeugs

15 Die Erfindung betrifft ein Steuergerät und ein Computerprogramm
zum Steuern eines Antriebsaggregates, insbesondere einer
Brennkraftmaschine eines Fahrzeugs.

Stand der Technik

20 Derartige Steuergeräte und zugehörige Computerprogramme sind im
Stand der Technik grundsätzlich bekannt. Figur 9 zeigt ein
derartiges bekanntes Steuergerät, wobei das Bezugszeichen 100a
die Hardware und das Bezugszeichen 100b die Software des
Steuergerätes 100 repräsentiert. Die Hardware des Steuergerätes
100 umfasst wenigstens einen Prozessor 100a-1 und wenigstens ein
Speicherelement 100a-2. Die Software 100b ist üblicherweise in
dem Speicherelement 100a-2 hinterlegt. Die Software 100b umfasst
im Stand der Technik üblicherweise eine Vielzahl von
Funktionseinheiten VF-1, EF-3, IS-2, HWE-1, HWE-3 und VF-3, die
zumindest vereinzelt zum Zwecke der Ansteuerung des
30 Antriebsaggregates 300 miteinander kommunizieren. Die
unmittelbare Steuerung des Antriebsaggregates 300 erfolgt mit
Hilfe einer zwischen das Steuergerät 100 und das
Antriebsaggregat 300 geschalteten Sensor-/Aktor-Konfiguration
200.

35 Bekannte Funktionseinheiten in der Software 100b für ein
Steuergerät 100 zum Steuern eines Antriebsaggregates sind
beispielsweise:

- 5 • Funktionseinheit Antrieb VF-1: Sie verwaltet die Quellen mechanischer Energie und stellt für das Antriebsaggregat ein Sollmoment zum Vortrieb des Fahrzeugs und zur Versorgung der Nebenaggregate, üblicherweise nach Vorgabe durch eine Funktionseinheit Fahrzeugkoordinator, zur Verfügung.
- 10 • Funktionseinheit Fahrzeugkoordinator VF-2: Sie trifft Entscheidungen über das Zusammenwirken verschiedener, auch anderer Funktionseinheiten. Beispielsweise entscheidet sie, welches Moment die Funktionseinheit Antrieb bei dem Antriebsaggregat einstellen soll, wenn verschiedene andere Funktionseinheiten jeweils unterschiedliche
15 Momentenbeträge bei dem Antriebsaggregat einzustellen fordern.
- 20 • Funktionseinheit Fahrzeugbewegung VF-3: Sie vergleicht eine aktuelle Bewegung des Fahrzeugs mit den Wünschen des Fahrers im Hinblick auf die Gewährleistung einer optimalen Fahrstabilität. Hierzu gehört zum Beispiel die Auswertung des Fahrerwunsches gemäß Aktionen am Gas- und Bremspedal sowie die Koordination dieses Fahrerwunsches mit Vorgaben von Sicherheitssystemen, wie elektronisches Stabilitätsprogramm ESP oder Antischlupfregelung ASR.
- Funktionseinheit Fahrzustandsgrößen VF-4: Sie verwaltet die Informationen über aktuelle Fahrzustände, zum Beispiel Stop & Go, Fahrt bergauf et cetera, unabhängig davon, welche Funktionseinheit diese Fahrzustände ermittelt.
- 30 • Funktionseinheit Motorkoordinator EF-1: Der Motorkoordinator hat die Aufgabe, alle Betriebszustände des Motors zu koordinieren und Informationen über den Motorbetriebszustand zur Verfügung zu stellen.
- 35 • Funktionseinheit Motormomentenstruktur EF-2: Diese Funktionseinheit hat die Aufgabe, die Momenten-Anforderungen von anderen Funktionseinheiten an den Motor zu ermitteln und zu koordinieren und eine resultierende Anforderung an die Momentenumsetzung zu ermitteln.
- Aggregat-Positions-Management EPM: Diese Funktionseinheit

hat die Aufgabe, eine Positions- und Drehzahlerfassung der Kurbel- und Nockenwelle des Antriebsaggregates durchzuführen.

- 5 • Funktionseinheit Dienstebibliotheken IS-1: Sie hat die Aufgabe, allgemeine, sehr häufig verwendete Funktionen bereitzustellen, die von unterschiedlichen Funktionseinheiten nachgefragt beziehungsweise benutzt werden. Sie bietet eine zentrale Bereitstellung dieser Funktionen mit dem Vorteil, dass diese nicht mehrfach dezentral bereitgestellt werden müssen.
- 10 • Funktionseinheit Ablaufsteuerung IS-2: Sie koordiniert die zeitliche Abarbeitung von Anforderungen unterschiedlicher Funktionseinheiten.
- 15 • Funktionseinheit Diagnosemanager IS-3: Diese Funktionseinheit übernimmt die Aufbereitung eines Fehlersignals, welches zum Beispiel einen Defekt in der Hardware 100a des Steuergerätes 100 repräsentiert. Die Aufbereitung besteht insbesondere in einer Entprellung des Fehlersignals und einer Abspeicherung desselben, damit zu
20 einem späteren Zeitpunkt eine Auswertung des Fehlersignals erfolgen kann.
- Funktionseinheit Überwachungskonzept IS-4: Diese Funktionseinheit dient insbesondere der Überwachung des Prozessors des Steuergerätes.
- Funktionseinheit Signalaufbereitung HWE-1: Sie führt eine Bereinigung eines analogen Sensorsignals nach dessen Digitalisierung durch im Hinblick auf unerwünschte Signalmodulationen, die in dem Steuergerät eventuell entstanden sein können.
- 30 • Funktionseinheit Gassystem EF-3: Sie hat die zentrale Aufgabe, Informationen über die aktuell für das Antriebsaggregat 300 zur Verfügung stehende Luftmasse zu liefern und im Rahmen ihrer Einflussmöglichkeiten auf das Antriebsaggregat eine gewünschte Soll-Luftmasse und/oder
35 die Abgasqualität zu überwachen oder einzustellen. Diese Funktionseinheit Gassystem wird insbesondere bei Dieselmotoren oder Benzinmotoren verwendet.

- Funktionseinheit Einspritzsystem EF-4: Sie stellt alle Funktionen bereit, welche zur Kraftstoffvorförderung, Einspritzdruckerzeugung und Einspritzung erforderlich sind.

5

Einzelne dieser bekannten Funktionseinheiten sind in Figur 9 innerhalb der Software 100b des Steuergerätes beispielhaft veranschaulicht. Sie wurden jedoch nicht alle gleichzeitig entwickelt und in der Software 100b implementiert, sondern sind erst im Laufe der Zeit im Zuge fortschreitender Entwicklung sukzessive zu der Software des Steuergerätes 100 hinzugefügt worden. Bei der Hinzufügung neuer Funktionseinheiten hat man bisher lediglich darauf geachtet, dass die neuen Funktionseinheiten mit allen anderen Funktionseinheiten, soweit erforderlich, kommunizieren konnten. Im Laufe der Zeit entstand so ein unübersichtliches Agglomerat von Schnittstellen zwischen den einzelnen Funktionseinheiten, was insbesondere einen Austausch bekannter Funktionseinheiten durch modifizierte Funktionseinheiten oder die weitere Hinzufügung von neuen Funktionseinheiten zunehmend schwieriger machte. Die Schwierigkeiten entstanden insbesondere deshalb, weil bestehende Abhängigkeiten zwischen einzelnen Funktionseinheiten bei einer Umgestaltung des Gesamtsystems oder auch nur von Teilen desselben nur noch sehr schwer überschaubar waren.

10

15

20

30

Ausgehend von diesem Stand der Technik ist es die Aufgabe der vorliegenden Erfindung, ein bekanntes Steuergerät und ein Computerprogramm zum Steuern eines Antriebsaggregates eines Fahrzeugs derart weiterzubilden, dass einzelne Teile des Steuergerätes und insbesondere von dessen Software unabhängig voneinander realisierbar und austauschbar sind.

35

Diese Aufgabe wird durch den Gegenstand des Patentanspruchs 1 gelöst. Demnach ist für ein oben beschriebenes bekanntes Steuergerät eine Modularisierung in Form von mindestens drei Modulen vorgesehen, wobei in einem ersten Modul diejenigen Funktionseinheiten zusammengefasst sind, die zur Beeinflussung des Antriebsaggregates im Ansprechen auf einen Benutzerwunsch

auf physikalischer Ebene dienen, in einem zweiten Modul diejenigen Funktionseinheiten zusammengefasst sind, welche eine individuelle Programmierung der Hardware des Steuergerätes in der Weise ermöglichen, dass die Hardware in die Lage versetzt wird, mit den Modulen des Steuergerätes zu kommunizieren und welche die Abarbeitung von Funktionen der Funktionseinheiten in den Modulen zeitlich koordinieren und in einem dritten Modul diejenigen Funktionseinheiten zusammengefasst sind, die eine individuelle Anpassung der verwendeten Sensor-/Aktor-Konfiguration an das Steuergerät in der Weise ermöglichen, dass zwischen den einzelnen Sensoren oder Aktoren der Konfiguration eine Kommunikation mit den übrigen Modulen des Steuergerätes möglich ist; und wobei zwischen den Modulen Modul-Schnittstellen für eine modul-übergreifende Kommunikation vorgesehen sind.

Das Antriebsaggregat im Sinne der Erfindung kann alternativ zu einer Brennkraftmaschine auch zum Beispiel einen Elektroantrieb oder einen Brennstoffzellenantrieb repräsentieren.

Benutzer im Sinne der Erfindung kann zum Beispiel der Fahrer des Fahrzeugs, der Gesetzgeber, ein Fahrzeugzulieferer oder ein Fahrzeughersteller sein.

Physikalische Ebene im Sinne der Erfindung bedeutet eine Abstraktion von Hardware-Spezifika. Dies bedeutet, dass in der physikalischen Ebene, bezogen auf die Schnittstelle des Sensors zu seiner Umgebung, lediglich physikalische Größen wie zum Beispiel die Drehzahl des Antriebsaggregates betrachtet werden, nicht jedoch deren hardwaremäßige Realisierung in Form eines elektronischen Signals mit einer hardware-spezifischen, die Drehzahl repräsentierenden Amplitude.

Vorteile der Erfindung

Der wesentliche Vorteil der beanspruchten Modularisierung besteht darin, dass die einzelnen Module einfach austauschbar und unabhängig voneinander realisierbar sind. Dabei wurde die Aufteilung der Funktionseinheiten auf die einzelnen Module so

gewählt, dass sie insbesondere aus Sicht der Fahrzeughersteller und aus physikalischer Sicht sinnvoll ist. Ein Austausch einzelner Module ist insbesondere deshalb besonders einfach, weil in den definierten Modul-Schnittstellen zwischen den einzelnen Modulen alle Abhängigkeiten zwischen den Funktionseinheiten in den verschiedenen Modulen berücksichtigt sind; darüber hinausgehende Abhängigkeiten können zwar noch auf Ebene der Funktionseinheiten existieren, sind aber auf Ebene der Module nicht mehr existent und brauchen deshalb bei einem Austausch der Module nicht mehr berücksichtigt zu werden.

Mit der vorgeschlagenen Modularisierung sind insbesondere Kosten- und Zeitersparnisse verbunden. Bei Verwendung einer anderen Steuergerätehardware oder einer anderen Sensor-/Aktor-Konfiguration braucht nun nicht mehr die gesamte Software des Steuergerätes ausgetauscht oder angepasst zu werden, sondern es ist vielmehr ausreichend, wenn nur das entsprechende Modul ausgetauscht wird.

Gemäß einem ersten Ausführungsbeispiel wird vorteilhafterweise vorgeschlagen, das erste Modul in eine Fahrzeug-Komponente und in eine Antriebsaggregate-Komponente zu unterteilen. Gemäß einem zweiten Ausführungsbeispiel wird vorgeschlagen, das zweite Modul in eine Infrastruktur-Komponente und in eine Hardwarekapsel sowie optional zusätzlich eventuell auch noch in eine Kommunikations-Komponente zu unterteilen.

Zusammenfassend zu den vorgenannten Ausführungsbeispielen ist anzumerken, dass die Unterteilung der einzelnen Module in Komponenten ähnlich wie bei den Modulen dazu dient, die Austauschbarkeit der Komponenten zu verbessern.

Vorteilhafterweise sind auch zwischen den Komponenten innerhalb eines Moduls Schnittstellen für einen schnellen Datenaustausch vorgesehen. Dagegen erfolgt eine Kommunikation zwischen Komponenten in unterschiedlichen Modulen über die Modul-Schnittstellen. Jede der Komponenten ist ihrerseits wiederum in beliebig viele Funktionseinheiten unterteilt. Schnittstellen

sind auch auf Ebene dieser Funktionseinheiten vorgesehen, so dass verschiedene Funktionseinheiten innerhalb einer Komponente miteinander kommunizieren können. Eine Kommunikation zwischen Funktionseinheiten in unterschiedlichen Komponenten erfolgt über die Komponentenschnittstellen und eine Kommunikation zwischen Funktionseinheiten in unterschiedlichen Modulen erfolgt über die Modul-Schnittstellen. Auch für die Schnittstellen zwischen den Funktionseinheiten und zwischen den Komponenten gilt das bereits oben für die Modul-Schnittstellen Gesagte, nämlich dass in diesen Schnittstellen alle notwendigen Abhängigkeiten zwischen den Funktionseinheiten oder Komponenten untereinander berücksichtigt sind und jeweils weitere Abhängigkeiten nicht berücksichtigt werden müssen. Daraus resultiert eine einfache Austauschbarkeit nicht nur von Modulen, sondern auch von Komponenten oder Funktionseinheiten, das heißt insbesondere eine einfache und unkomplizierte Anpassung an Kundenwünsche oder neue Technologien.

Eine nähere Beschreibung der Aufgaben der einzelnen Komponenten und der in den einzelnen Komponenten zusammengefassten Funktionseinheiten und Elemente ist Gegenstand der Unteransprüche.

Gemäß einer vorteilhaften Ausbildung sind die Funktionseinheiten, die Komponenten und/oder die Module sowie deren jeweilige Schnittstellen zumindest teilweise als Computerprogramm ausgebildet. Die Ausbildung als Computerprogramm erlaubt eine flexible Umsetzung von Änderungswünschen, ohne dass eine Änderung der Hardware des Steuergerätes vorgenommen werden muss.

Die oben genannte Aufgabe wird weiterhin durch ein Computerprogramm für das beschriebene und beanspruchte Steuerprogramm gelöst. Die Vorteile dieses Computerprogramms entsprechen den oben mit Bezug auf das Steuerprogramm erwähnten Vorteilen.

Weitere Vorteile ergeben sich aus der nachfolgenden Beschreibung.

Zeichnungen

Der Beschreibung sind insgesamt neun Figuren beigelegt, wobei

- 5 Figur 1 eine erfindungsgemäße Modularchitektur für ein Steuergerät;
- 10 Figur 2 eine erfindungsgemäße Unterteilung der Module in Komponenten;
- 10 Figur 3 eine erfindungsgemäße Zuordnung von Funktionseinheiten zu einer Fahrzeug-Komponente und einer Antriebsaggregate-Komponente;
- 15 Figur 4 die erfindungsgemäße Zuordnung von Funktionseinheiten zu einer Infrastruktur-Komponente und einer Hardwarekapsel-Komponente;
- 20 Figur 5 eine Funktionseinheit für ein Gerätetreiber-Element;
- 20 Figur 6 eine Veranschaulichung von Schnittstellen zwischen den Modulen;
- 25 Figur 7 eine Schnittstelle zwischen zwei Funktionseinheiten in unterschiedlichen Komponenten;
- 25 Figur 8 ein Beispiel für einen Datenfluss zwischen den Modulen; und
- 30 Figur 9 den Aufbau eines Steuergerätes gemäß dem Stand der Technik
- 35 Figur 10 die Einbettung einer erfindungsgemäßen Schnittstellenarchitektur in die Modularchitektur
- 35 Figur 11 die Schichtendarstellung einer ausgewählten Schnittstelle
- zeigt.

Beschreibung der Ausführungsbeispiele

5 Es folgt eine detaillierte Beschreibung von
Ausführungsbeispielen der Erfindung unter Bezugnahme auf die
Figuren.

10 Figur 1 zeigt unter Bezugnahme auf Figur 9 eine erfindungsgemäße
Modularisierung von Funktionseinheiten in einem Steuergerät 100
zum Steuern eines Antriebsaggregates 300, insbesondere einer
Brennkraftmaschine, eines Fahrzeugs. Im Unterschied zu Figur 9
ist in Figur 1 lediglich die Struktur des Inhaltes des
Speicherelementes 100a-2 dargestellt; insbesondere die Sensor-
/Aktor-Konfiguration 200 und das Antriebsaggregat 300 sind nicht
15 Gegenstand der Erfindung und werden deshalb nachfolgend nicht
weiter beschrieben.

20 Erfindungsgemäß wird vorgeschlagen, die Funktionseinheiten in
dem Steuergerät 100 in vier separate Module zu gruppieren.

In einem ersten Modul ASW werden diejenigen Funktionseinheiten
zusammengefasst, die zur Beeinflussung des Antriebsaggregates
300 im Ansprechen auf einen Benutzerwunsch auf physikalischer
Ebene dienen.

25 In einem zweiten Modul CO werden zum einen diejenigen
Funktionseinheiten zusammengefasst, welche eine individuelle
Programmierung der Hardware des Steuergerätes 100 in der Weise
ermöglichen, dass die Hardware in die Lage versetzt wird, mit
30 den Modulen des Steuergerätes 100 zu kommunizieren. Weiterhin
umfasst das zweite Modul CO diejenigen Funktionseinheiten,
welche eine Abarbeitung von Funktionen der Funktionseinheiten in
den Modulen ASW, CO, DE, CD zeitlich koordinieren. Darüber
hinaus kann das zweite Modul CO auch solche Funktionseinheiten
35 aufweisen, welche eine Kommunikation des Moduls ASW und/oder
eines dritten Moduls DE mit anderen Steuergeräten ermöglichen.

5 In einem dritten Modul DE werden diejenigen Funktionseinheiten zusammengefasst, die eine individuelle Anpassung der verwendeten Sensor-/Aktor-Konfiguration an das Steuergerät 100 in der Weise ermöglichen, dass zwischen den einzelnen Sensoren oder Aktoren der Konfiguration eine Kommunikation mit den übrigen Modulen des Steuergerätes 100 möglich ist.

10 Schließlich werden in einem vierten Modul CD diejenigen Funktionseinheiten zusammengefasst, die eine direkte Ansteuerung von komplexen Sensor-/Aktor-Konfigurationen mit komplexen Schnittstellen zu dem Steuergerät 100 durch das erste Modul ermöglichen. Diese speziellen Sensor-/Aktor-Konfigurationen sind von den zuvor erwähnten nicht-speziellen Sensor-/Aktor-Konfigurationen zu unterscheiden. Im Unterschied zu nicht-
15 spezifischen Konfigurationen, bei denen eine Kommunikation mit dem ersten Modul nur über das zweite und das dritte Modul ist, erfolgt bei den speziellen Konfigurationen eine Kommunikation mit dem ersten Modul direkt über das vierte Modul.

20 Zwischen den Modulen ASW, CO, DE, CD sind jeweils Modul-Schnittstellen M1, M2, M3, M4, M5 und M6 vorgesehen, die zum einen eine Kommunikation der Module untereinander, aber auch den Austausch einzelner Module ermöglichen.

25 Figur 2 zeigt eine erfindungsgemäße Gruppierung von Komponenten innerhalb der zuvor beschriebenen Module ASW, CO und CD. Das erste Modul ASW repräsentiert im Wesentlichen anwender- bzw. benutzerorientierte Funktionen. Im Hinblick auf geplante Anwendungsfälle, bei denen unterschiedliche Typen von
30 Antriebsaggregaten durch das Steuergerät angesteuert werden sollen, ist es sinnvoll, dieses erste Modul ASW in eine Fahrzeug-Komponente VF und eine Antriebsaggregate-Komponente EF zu unterteilen.

35 Die Fahrzeug-Komponente umfasst vorzugsweise diejenigen Funktionseinheiten, die nicht spezifisch für einen bestimmten Typ von Antriebsaggregat 300 sind. Dabei handelt es sich insbesondere um die Funktionseinheiten Antrieb VF-1,

Fahrzeugkoordinator VF-2, Fahrzeugbewegung VF-3 oder Fahrzustandsgrößen VF-4, wie sie einleitend beschrieben wurden.

Demgegenüber sind in der Antriebsaggregate-Komponente EF vorzugsweise alle diejenigen Funktionseinheiten zusammengefasst, die für den verwendeten Typ von Antriebsaggregat 300 spezifisch sind. Dabei handelt es sich vorzugsweise um die Funktionseinheiten Motorkoordinator EF-1, Motormomentenstruktur EF-2, Gassystem EF-3 oder Einspritzsystem EF-4, wie oben unter Bezugnahme auf Figur 9 beschrieben. Bei einem Wechsel des von dem Steuergerät 100 anzusteuernenden Antriebsaggregates 300 ist es dann nicht mehr erforderlich, das komplette erste ASW-Modul auszutauschen, sondern es genügt vorteilhafterweise, wenn lediglich die Antriebsaggregate-Komponente EF ausgetauscht wird.

Ähnlich wie bei dem ersten Modul empfiehlt es sich bei dem zweiten Modul zum Beispiel im Hinblick auf einen anderen zu verwendenden Prozessor in dem Steuergerät 100, das zweite Modul in eine Infrastruktur-Komponente IS und in eine Hardwarekapsel-Komponente HWE zu unterteilen. In der Infrastruktur-Komponente IS sind vorzugsweise alle diejenigen Funktionseinheiten zusammengefasst, welche grundlegende Dienste anbieten oder repräsentieren, auf die andere Funktionseinheiten zugreifen können. Dabei handelt es sich vorzugsweise um die Funktionseinheiten Dienstbibliotheken IS-1, Ablaufsteuerung IS-2, Diagnosemanager IS-3 und Überwachungskonzept IS-4, wie ebenfalls oben beschrieben. Die Dienste werden in der Infrastruktur-Komponente an zentraler Stelle bereitgestellt und müssen deshalb nicht dezentral in mehrfacher Ausfertigung vorliegen und unnötig Speicherplatz beanspruchen.

In der Hardwarekapsel-Komponente HWE sind alle diejenigen Funktionseinheiten zusammengefasst, welche eine individuelle Programmierung der Hardware 100a des Steuergerätes 100 in der Weise ermöglichen, dass die Hardware 100a in die Lage versetzt wird, mit den Modulen ASW, CO, DE oder CD des Steuergeräts 100 zu kommunizieren. So ist es bei einem Austausch des in dem Steuergerät verwendeten Prozessors durch einen Prozessor anderen

Typs lediglich erforderlich, die Hardwarekapsel-Komponente HWE und nicht mehr das gesamte zweite Modul des Steuergerätes auszutauschen.

- 5 Neben der genannten Infrastruktur-Komponente und der Hardwarekapsel-Komponente kann das zweite Modul CO weiterhin eine Kommunikations-Komponente COM aufweisen, in welcher diejenigen Funktionseinheiten zusammengefasst sind, welche eine Kommunikation mit anderen Steuergeräten ermöglichen.

10

Neben den bereits oben erwähnten Modul-Schnittstellen M1 ... M6 auf Modulebene sind auch Komponentenschnittstellen auf Ebene der Komponenten innerhalb der einzelnen Module vorgesehen. Diese Komponentenschnittstellen K0 ... K3 ermöglichen einen

- 15 Datenaustausch zwischen zumindest einzelnen der Komponenten innerhalb eines Moduls. So besteht eine Schnittstelle K0 in dem ersten Modul zwischen der Fahrzeug-Komponente VF und der Antriebsaggregate-Komponente EF. Innerhalb des zweiten Moduls besteht eine Komponentenschnittstelle K1 zwischen der

- 20 Infrastruktur-Komponente IS und der Kommunikations-Komponente COM, eine zweite Schnittstelle K2 zwischen der Infrastruktur-Komponente IS und der Hardwarekapsel-Komponente HWE und eine dritte Schnittstelle K3 zwischen der Hardwarekapsel-Komponente HWE und der Kommunikations-Komponente COM.

25

In Figur 3 ist die bereits erwähnte Zuordnung von Funktionseinheiten zu der Fahrzeug-Komponente VF und zu der Antriebsaggregate-Komponente sowie die Schnittstelle K0 zwischen den beiden Komponenten dargestellt. Darüber hinaus ist

- 30 ersichtlich, dass auch die Funktionseinheiten innerhalb einer Komponente über Funktionseinheiten-Schnittstellen F_i mit $i = 1 - 9$ untereinander kommunizieren können. Eine Kommunikation zwischen Funktionseinheiten, die unterschiedlichen Komponenten innerhalb desselben Moduls zugeordnet sind, erfolgt über die
- 35 besagte Komponentenschnittstelle, hier K0.

Figur 4 zeigt die bereits beschriebene Zuordnung von Funktionseinheiten zu der Infrastruktur-Komponente IS und der

Hardwarekapsel-Komponente HWE innerhalb des zweiten Moduls. Die bereits unter Bezugnahme auf Figur 3 gemachten Aussagen zu den Schnittstellen zwischen den Funktionseinheiten und den Komponenten gelten hier entsprechend. Alternativ oder zusätzlich zu der in den Figuren 3 und 4 gezeigten Ausbildung der Funktionseinheiten-Schnittstellen zwischen den Funktionseinheiten innerhalb einer Komponente können diese Schnittstellen auch derart ausgebildet sein, dass eine direkte Verbindung zu jeder Funktionseinheit innerhalb einer Komponente besteht. Für eine Kommunikation zwischen beispielsweise der Funktionseinheit Dienstbibliotheken IS-1 und der Funktionseinheit Diagnosemanager IS-3 wäre es dann nicht erforderlich, über die Funktionseinheit Ablaufsteuerung IS-2 zu kommunizieren, sondern es stünde eine direkte Verbindung zur Verfügung.

Für die Funktionseinheiten HWE-1...N der HWE ist es vorteilhaft, diese jeweils in ein Prozessorebenenelement und ein Hardwareabstraktionsebenenelement zu unterteilen. Diese Unterteilung bietet den Vorteil, dass bei einem gewünschten Austausch des Prozessors nicht die gesamte Hardware-Kapsel-Komponente HWE, sondern nur die Prozessorebenenelemente, dann allerdings bei allen Funktionseinheiten der Hardwarekapsel-Komponente HWE ausgetauscht werden müssen. Analog ist es dann bei einem gewünschten Austausch der Peripheriehardware des Steuergerätes, das ist die Hardware des Steuergerätes außer dem Prozessor, ebenfalls nicht mehr erforderlich, die gesamte Hardwarekapsel-Komponente HWE auszutauschen, sondern es genügt ein Austausch der Hardwareabstraktionsebenenelemente bei allen Funktionseinheiten der HWE.

Eine der Hardwarekapsel-Komponente HWE zugeordnete Funktionseinheit ist die Funktionseinheit Signalaufbereitung HWE-1, die einleitend beschrieben wurde. Für die Funktionseinheit Signalaufbereitung HWE-1 übernimmt das Prozessorebenenelement HWE-1-1 die Signalaufbereitung insoweit als sie sich auf den verwendeten Prozessortyp bezieht und das Hardwareabstraktionsebenenelement HWE-1-2 die Signalaufbereitung

insoweit als sie sich auf die im Steuergerät verwendete Peripheriehardware bezieht.

5 Schnittstellen Ek sind weiterhin vorgesehen für eine Kommunikation der Prozessorebenelemente HWE-1-1 und der Hardwareabstraktionsebenelemente HWE-1-2 untereinander und/oder mit ihnen übergeordneten Funktionseinheiten HWE-1 in der HWE.

10 Figur 5 veranschaulicht die bereits erwähnte vorteilhafte Unterteilung des vierten Moduls CD in mehrere Funktionseinheiten; diese werden in Bezug auf das vierte Modul auch Komplexgerätetreiber CD-i mit $i = 1 \dots N$ genannt. Jeder dieser Komplexgerätetreiber wird seinerseits vorteilhafterweise in ein Gerätetreiber-Element CD-GT und in ein Hardwaretreiber-Element CD-HW unterteilt. Die Elemente CD-GT sind über die Schnittstelle M3 an das erste Modul ASW angeschlossen. Über diese Schnittstelle M3 werden Softwaresignale übertragen, welche eine physikalische Größe, wie zum Beispiel die Einspritzmenge oder Drehzahl repräsentieren. Wenn die Gerätetreiber-Elemente CD-GT ein solches Softwaresignal von dem ersten Modul ASW empfangen, wandeln sie dieses in ein anderes Softwaresignal um, welches spezifisch ist für einen anzusteuernenden Aktor, aber noch unspezifisch für die Steuergeräte-Hardware ist. Im umgekehrten Fall, wenn die Gerätetreiber-Elemente CD-GT ein derartiges anderes Softwaresignal von einem Hardwaretreiber-Element CD-HW empfangen, wandeln sie dieses in ein Softwaresignal um, welches nur eine physikalische Größe repräsentiert, aber nicht mehr spezifisch ist für einen Sensor, von dem es ursprünglich erzeugt wurde. Die Hardwaretreiber-Elemente CD-HW dienen zum direkten Anbinden der speziellen Sensor-/Aktor-Konfiguration 200 an die verwendete Steuergeräte-Hardware, insbesondere an den verwendeten Prozessor 100a-1. Zwischen den Gerätetreiber-Elementen CD-GT und den zugehörigen weiteren Hardwaretreiber-Elementen CD-HW sind jeweils Schnittstellen Ei mit $i = 1 \dots N$ vorgesehen.

Einer der Komplexgerätetreiber ist die einleitend beschriebene Funktionseinheit Aggregate-Positions-Management EPM. Ihr

Gerätetreiber-Element und ihr Hardwaretreiber-Element erhalten zum Zwecke einer eindeutigen Zuordnung die Bezugszeichen CD_{EPM-GT} und CD_{EPM-HW} .

- 5 Figur 6 veranschaulicht beispielhaft die Modulschnittstellen M2 und M4. Die Pfeile repräsentieren jeweils die durch die Schnittstellen realisierten Abhängigkeiten zwischen den gezeigten Modulen oder deren Komponenten. Dabei bedeutet die Aussage, dass ein Modul abhängig von einem weiteren Modul oder
10 eine Komponente abhängig von einer weiteren Komponente ist, dass dieses Modul auf das weitere Modul oder diese Komponente auf die weitere Komponente zugreift beziehungsweise dem weiteren Modul oder der weiteren Komponente Daten zur Verarbeitung in Auftrag gibt. Die die Abhängigkeiten repräsentierenden Pfeile stellen
15 nicht zwangsläufig auch die entsprechenden Datenflussrichtungen dar; diese werden beispielhaft weiter unten mit Bezugnahme auf Figur 8 erläutert. Aus Figur 6 ist zu erkennen, dass zwischen dem dritten Modul DE und dem ersten Modul ASW eine wechselseitige Abhängigkeit besteht. Die Abhängigkeit des ersten
20 Moduls ASW von dem dritten Modul DE, repräsentiert durch den Pfeil, der von dem zweiten Modul zu dem ersten Modul orientiert ist, resultiert aus der Tatsache, dass das erste Modul darauf angewiesen ist, dass das dritte Modul DE zum Beispiel ein Sensorsignal und/oder ein zugeordnetes Qualitätssignal, welches
25 eine Information über die Qualität des Sensorsignals repräsentiert, für das erste Modul ASW bereitstellt.

- Wie aus Figur 6 weiterhin zu erkennen ist, realisiert die vierte Modul-Schnittstelle M4 einseitige Abhängigkeiten der
30 Infrastruktur-Komponente und der Hardwarekapsel-Komponente gegenüber dem dritten Modul DE und eine wechselseitige Abhängigkeit zwischen der Kommunikations-Komponente COM gegenüber dem dritten Modul DE. Alle drei genannten Komponenten sind dem zweiten Modul CO zugeordnet. Die einseitige
35 Abhängigkeit der Infrastruktur-Komponente und der Hardwarekapsel-Komponente bedeutet, dass auf das dritte Modul DE auf diese Komponenten zugreift, um dort Daten verarbeiten zu lassen. Analoge Abhängigkeitsbezeichnungen, wie sie soeben für

die Schnittstelle M4 zwischen dem zweiten Modul CO und dem dritten Modul DE beschrieben wurden, bestehen auch für die Schnittstelle M6 zwischen dem zweiten Modul CO und dem vierten Modul CD.

5

Figur 7 zeigt ein Beispiel für eine komponentenübergreifende Kommunikation zwischen zwei Funktionseinheiten in unterschiedlichen Komponenten desselben Moduls. Genauer gesagt veranschaulicht Figur 7 eine Kommunikation zwischen der Funktionseinheit Signalaufbereitung HWE-1 innerhalb der Hardwarekapsel-Komponente HWE und der Funktionseinheit Diagnosemanager IS-3 innerhalb der Infrastruktur-Komponente IS. Hier ist die Funktionseinheit Signalaufbereitung abhängig von der Funktionseinheit Diagnosemanager. Dies kann zum Beispiel bedeuten, dass die Funktionseinheit Signalaufbereitung HWE-1 ein Signal an die Funktionseinheit Diagnosemanager IS-3 sendet mit dem Auftrag um Bearbeitung. Das gesendete Signal repräsentiert zum Beispiel einen Defekt in der dem Steuergerät 100 zugeordneten Hardware 100a. Mit der Übersendung des Signals beauftragt dann die Funktionseinheit Signalaufbereitung HWE-1 die Funktionseinheit Diagnosemanager, dieses Signal aufzubereiten, das heißt zum Beispiel zu entprellen, so dass eine fehlerfreie Auswertung dieses Signals möglich wird. Neben der Signalaufbereitung kommt dann der Funktionseinheit Diagnosemanager zum Beispiel auch die Aufgabe zu, dieses Signal abzuspeichern, so dass es zu einem späteren Zeitpunkt, zum Beispiel in einer Kfz-Werkstatt, zwecks Fehlerdiagnose ausgelesen werden kann.

10

15

20

25

Figur 8 veranschaulicht schließlich das Zusammenwirken der einzelnen Module innerhalb der Steuergeräte-Software 100b anhand eines beispielhaften Verlaufs von Signalen. Eine derartige Signalverlaufsbetrachtung ist eine alternative Möglichkeit zur Beschreibung der Funktionen einzelner Schnittstellen, insbesondere zwischen den Modulen, im Vergleich zu der oben bereits erwähnten Abhängigkeitsbetrachtung. Beide Betrachtungsweisen sind keineswegs widersprüchlich zueinander,

30

35

sondern beschreiben lediglich die Funktionen einer Schnittstelle auf unterschiedliche Weise.

5 Konkret ist in Figur 8 derjenige Signalverlauf dargestellt, der sich ergibt, wenn der Fahrer eines Fahrzeugs, in welches die Brennkraftmaschine 300 mit dem Steuergerät 100 eingebaut ist, das Gaspedal betätigt. Das Gaspedal beziehungsweise ein daran befestigter Positionsdetektor wird in Figur 8 durch das Bezugszeichen 200a als Sensor der Sensor-/Aktor-Konfiguration 10 200 repräsentiert. Ein Signal S1, welches die veränderte Stellung des Gaspedals repräsentiert, wird von dem Gaspedal zunächst in die Hardware 100a des Steuergerätes 100 eingegeben. Innerhalb der Steuergeräte-Hardware 100a wird das reale elektrische Sensorsignal umgewandelt in ein Softwaresignal, 15 welches zunächst noch steuergerätespezifisch ist. Dieses Softwaresignal S2 verlässt alsdann die Steuergeräte-Hardware 100a und wird in das zweite Modul CO, genauer gesagt in dessen Hardwarekapsel-Komponente HWE eingespeist. Dort erfährt das Softwaresignal eine Aufbereitung in der Weise, dass die in ihm 20 noch enthaltenen Einflüsse der Steuergeräte-Hardware 100a aus ihm entfernt werden. Am Ausgang der Hardwarekapsel-Komponente HWE liegt dann ein bereinigtes Softwaresignal an, welches insbesondere keine Prozessorspezifika mehr enthält. Es repräsentiert jedoch nach wie vor noch die Charakteristika des 25 ursprünglichen elektrischen Signals, nämlich die veränderte Gaspedalstellung, zum Beispiel in Form einer Amplitude von 3 V. Dieses bereinigte Signal S3 wird dann über die Modul-Schnittstelle M4 dem dritten Modul DE zugeführt. In dem dritten Modul DE wird das bereinigte Sensorsignal S3 in der Weise 30 aufbereitet, dass es auf eine physikalische Ebene bezogen auf die Schnittstelle des Sensors zu seiner Umgebung umgewandelt wird. Beispielsweise kann dies bedeuten, dass das physikalische Signal S4 am Ausgang des dritten Moduls DE die durch das Signal S3 am Eingang des dritten Moduls in Form der 3 V Amplitude 35 repräsentierte veränderte Stellung des Gaspedals als Ist-Größe zum Beispiel in Form von 75 % des möglichen Drehwinkels des Gaspedals repräsentiert. Das physikalische Signal S4 ist Teil der Schnittstelle M2 zwischen dem ersten Modul ASW und dem

5 dritten Modul DE. Es führt von dem dritten Modul direkt in das
erste Modul ASW, genauer gesagt in dessen Fahrzeug-Komponente
VF. Die Fahrzeug-Komponente VF interpretiert das physikalische
Eingangssignal S4 als Momentenanforderung für die
10 Brennkraftmaschine 300. Sie koordiniert diese
Momentenanforderung von dem dritten Modul mit eventuell
vorhandenen anderen Momentenanforderungen, die von anderen
Modulen, Komponenten oder Funktionseinheiten an die
Brennkraftmaschine gestellt werden, um letztendlich ein
15 resultierendes Solldrehmoment für die Brennkraftmaschine in Form
des Signals S5 über die Komponentenschnittstelle K0 an die
Antriebsaggregate-Komponente EF auszugeben.

15 Die Antriebsaggregate-Komponente EF führt dann eine Umwandlung
des empfangenen Solldrehmomentes in von dem Typ der
Brennkraftmaschine abhängige physikalische Größen durch. Handelt
es sich bei der Brennkraftmaschine 300 beispielsweise um eine
Dieselmaschine, so erzeugt die Aggregate-Komponente EF ein
20 erstes Stellgrößensignal S6, welches den physikalischen Soll-
Einspritzdruck repräsentiert, der zur Einstellung des
erforderlichen Solldrehmomentes erforderlich ist, sowie ein
zweites Stellgrößensignal S10, welches die zur Einstellung des
vorgegebenen Solldrehmomentes erforderliche Sollkraftstoffmenge
auf physikalischer Ebene repräsentiert. Je nachdem, ob die
25 generierte Stellgröße für einen normalen Aktor oder einen
speziellen Aktor mit komplexer Schnittstelle vorgesehen ist,
gibt die Antriebsaggregate-Komponente EF die Stellgröße an das
dritte Modul DE oder an das vierte Modul CD aus. In dem bisher
beschriebenen Beispiel ist die erste Stellgröße S6 zur
30 Ansteuerung eines Druckregelventils 200b2 vorgesehen, bei dem es
sich um einen normalen Aktor ohne komplexe Schnittstelle
handelt. Deshalb gibt die Antriebsaggregate-Komponente EF das
erste Stellgrößensignal S6 über die Schnittstelle M2 an das
dritte Modul DE aus. Das dritte Modul wandelt die eingegebene
35 physikalische Stellgröße S6 (Softwaresignal) um in ein
Softwaresignal, welches den physikalischen Solldruck in Form
eines für die Steuergeräte-Hardware 100a unspezifischen
elektrischen Signals S7 repräsentiert. Dieses Signal S7 wird

über die Modul-Schnittstelle M4 an das zweite Modul CO
ausgegeben, wo es von dessen Hardwarekapsel-Komponente HWE in
ein Softwaresignal umgewandelt wird, welches ein für die
Steuergeräte-Hardware 100a spezifisches elektrisches Signal
repräsentiert. Bei dieser Umwandlung kann es sich beispielsweise
um eine an die Anforderungen der Steuergeräte-Hardware
angepasste Quantisierung handeln. Das von der Hardwarekapsel-
Komponente HWE erzeugte und ausgegebene Signal S8 wird der
Steuergeräte-Hardware 100a zugeführt, welche dieses Signal in
ein reales elektrisches Signal zur Ansteuerung des
Druckregelventils 200b2 als Aktor umwandelt.

Parallel zu der Bearbeitung des ersten Stellgrößensignals S6
erfolgt in dem beschriebenen Beispiel die Bearbeitung des
zweiten Stellgrößensignals S10 durch das vierte Modul CD,
nachdem es dorthin über die Modul-Schnittstelle M3 übertragen
wurde. Das vierte Modul wandelt das Signal S10, welches die zur
Realisierung des angeforderten Solldrehmomentes erforderliche
Sollkraftstoffmenge in physikalischer Form repräsentiert, in ein
Softwaresignal um, welches ein für die Steuergeräte-Hardware
100a spezifisches Signal repräsentiert. Insofern übernimmt das
vierte Modul CD gleichzeitig die Funktion des dritten Moduls und
der Hardwarekapsel-Komponente für spezielle Aktoren mit
komplexerer Schnittstelle, wie es ein Einspritzventil 200b1 zur
Einstellung der Kraftstoffmenge darstellt. Das beschriebene, von
dem vierten Modul CD ausgegebene Softwaresignal S11 wird dann
ebenfalls der Steuergeräte-Hardware 100a zugeführt, damit diese
das empfangene Softwaresignal in ein reales elektrisches Signal
S12 zur Ansteuerung des Einspritzventils 200b1 als Aktor
umwandelt. Die so erfolgte Ansteuerung des Druckregelventils
200b2 und des Einspritzventils 200b1 bewirken zusammen eine
Veränderung des Drehmomentes der Brennkraftmaschine 300 im
Hinblick auf den durch die veränderte Gaspedalstellung
repräsentierten Fahrerwunsch beziehungsweise auf das diesen
Fahrerwunsch repräsentierende Solldrehmoment.

Würde es sich bei der Brennkraftmaschine 300 nicht um eine
Dieselmaschine, sondern um eine Benzinmaschine handeln, so
würden von der Antriebsaggregate-Komponente EF drei

Stellgrößensignale erzeugt. Das erste Stellgrößensignal S6, welches auf das dritte Modul DE ausgegeben wird, repräsentiert einen Sollwert für die einzustellende Luftmasse und das zweite auf das vierte Modul ausgegebene Stellgrößensignal S10 repräsentiert die zur Realisierung des Solldrehmomentes erforderliche Sollkraftstoffmenge. Weiterhin wird ein drittes Stellgrößensignal S13, ebenfalls von der Antriebsaggregate-Komponente EF an das vierte Modul CD ausgegeben, wobei das Stellgrößensignal S13 den Zündzeitpunkt für die Zündkerzen der Brennkraftmaschine definiert.

Das erste Stellgrößensignal S6 dient nach einer Umwandlung in die Signale S7, S8 und S9 zur Ansteuerung der Drosselklappe, das zweite Stellgrößensignal S10 dient nach einer Umwandlung in die Signale S11 und S12 wiederum zur Ansteuerung des Einspritzventils und das dritte Stellgrößensignal S13 dient nach einer Umwandlung in die Signale S14 und S15 zur Ansteuerung einer Zündkerze 200b3. Die in Figur 8 innerhalb der Module oder Komponenten gezeichneten gestrichelten Linien veranschaulichen lediglich die Zuordnungen von Eingangssignal zu Ausgangssignal. Sie schließen eine Bearbeitung der Eingangssignale innerhalb der Module oder Komponenten ausdrücklich nicht aus.

Die Funktionseinheiten, Komponenten oder Module werden vorzugsweise als Computerprogramme realisiert. Dann ist es möglich, diese Computerprogramme, gegebenenfalls zusammen mit weiteren Computerprogrammen zur Steuerung der Antriebsaggregate, auf einem computerlesbaren Datenträger abzuspeichern. Dabei kann es sich um eine Diskette, eine Compact Disc, einen sogenannten Flash-Memory oder dergleichen handeln. Die auf dem Datenträger abgespeicherte Software kann dann als Produkt an einen Kunden verkauft werden. Außerdem ist es im Fall einer Softwarerealisierung möglich, das Computerprogramm, wiederum gegebenenfalls zusammen mit weiteren Computerprogrammen, auch ohne die Zuhilfenahme eines Datenträgers über ein elektronisches Kommunikationsnetzwerk als Produkt an einen Kunden zu übertragen und auf diese Weise zu verkaufen. Bei dem Kommunikationsnetzwerk kann es sich zum Beispiel um das Internet handeln.

In einer speziellen Erweiterung der bisher beschriebenen Ausführungsformen wird nun konkret auf die Konfiguration und Zuordnung der Ein- und Ausgangssignale der Module, insbesondere DE und CO sowie auf deren Beschreibung abgehoben. Dabei wird ein Zwischenmodul bzw. eine Zwischenschicht SZS eingeführt, welche eine Zuordnung der Signale zwischen den Modulen, insbesondere CO und DE ermöglicht. Die ist in der Figur 10 dargestellt. Darin sind wieder die Module CO, CD, DE, ASW wie bereits beschrieben dargestellt. Zwischen den Modulen CO und DE wird nun eine konkrete Schnittstelle zur Strukturierung von Konfigurationsdaten, die so genannte Konfigurationsschnittstelle KSS gezeigt. Diese enthält als wesentlichen Anteil ein Routing Modul zur Signalzuordnung also eine Signal-Zuordnung-Schicht SZS. Diese SZS ist der Hardwarenäheren Schicht, also dem Modul CO zugeordnet. Des weiteren ist ein Anforderungsmodul oder eine Anforderungsschicht AS dem Modul DE zugeordnet. Durch diese SZS bzw. die gesamte KSS kann nun eine allgemeine und flexible Signalzuordnung erfolgen, so dass das Modul CO und das Modul DE unabhängig von ihren miteinander ausgetauschten Ein- und Ausgangssignalen bzw. deren konkreter Anordnung konzipiert werden können, da die korrekte Signalzuordnung in einer Zwischen-Schicht, dem Modul SZS erfolgt.

Bei einer ausgewählten Darstellung erfolgt die Konfiguration von Modulen vorteilhafter Weise über eine XML Beschreibung der Konfigurationsparameter, die durch Konfigurationsgeneratoren in entsprechende *.c- und *.h-Dateien übersetzt werden. Dieses Dokument beschreibt die Konfiguration des Software-Paketes Digital Input/Output (DIO) über XML-basierte Textdateien. In diesen XML-Dateien werden u.a. die Eigenschaften der Signalklasse DIO definiert. Die Beschreibung der DIO-Eigenschaften wird auf verschiedenen Arbeitsebenen durchgeführt. Einerseits werden DIO-Signale von der projekt-unabhängigen Device Encapsulation (DE) mit den Parametern Richtung, Initialisierungswert und Applizierbarkeit angefordert (DIO_SIGNAL_REQUEST), andererseits müssen diese Signale auf die projektspezifische Anwendung projiziert werden, d.h. die Signale

müssen auf die entsprechende Hardware geroutet werden (DIO_SIGNAL_ROUTING). Angeforderte Hardware Pins müssen gleichfalls konfiguriert werden (DIO_SIGNAL_IMPLEMENT).

5 Beschreibung auf unterschiedlichen Konfigurationsebenen.

Bei der Beschreibung der digitalen Ein- und Ausgangssignale wird zwischen Device Encapsulation (DE), Hardware Abstraction Layer (HAL= SZS) und Hardware Configuration (Modul CO) unterschieden.

10 Diese Anordnung findet sich aber auch zwischen weiteren Schichten bzw. ist zwischen den übrigen Modul ebenso einsetzbar. Optional zum Beispiel zwischen ASW und CO als KSS3 oder ASW und DE als KSS4 oder ASW und CD als KSS5 sowie CD und DE als KSS2.

15 Dabei ist immer wie bei KSS das Ausgangsmodul wie hier CO als Hardware Configuration, ein Ziel-Modul, wie DE bei KSS mit einem Anforderungsmodul bzw. einer Anforderungsschicht AS sowie eine Zuordnungsschicht bzw. ein Routing-Modul SZS enthalten. Diese Grundstruktur ist ebenfalls auf die übrigen Konfigurationsschnittstellen KSS2 bis KSS5 übertragbar. Im
20 weiteren wird nur die KSS, also CO und DE betreffend beschrieben, wobei dadurch die Anwendbarkeit die übrigen Schnittstellen KSS2-KSS5 als ebenso veranschaulicht angesehen wird.

25 Die relevanten Konfigurationsparameter für die Komponententreiber der DE können abstrakt und projektunabhängig dargestellt werden. Bei digitalen Ein- und Ausgangssignalen sind auf dieser Ebene beispielsweise der Signalnamen, die Richtung und der Initialisierungswert von Interesse. Dabei spielt es aus
30 Sicht der Komponententreiber keine Rolle (die Einhaltung der Rahmenbedingungen vorausgesetzt), auf welcher Hardware der Komponententreiber später laufen soll.

35 Die Konfiguration der HAL umfasst das Routing des Signalnamens auf die entsprechende Hardware. In dieser Schicht wird dem Signal ein Generischer Hardwarepinnamen zugewiesen.

Die Hardware-spezifischen Konfigurationseinstellungen werden auf unterster Ebene vorgenommen. Die Konfigurationsparameter dieser Ebene beziehen sich auf das entsprechende Modul bzw. auf die Signalklasse und lassen sich bereits nicht mehr Projekt- und Hardware-unabhängig betrachten. Die Zuordnung der Signale geschieht über die Angabe des Hardwaremoduls (z.B. ADC, GPIO oder CY310, CY100, etc.) mit dem gewünschten Port bzw. Pin auf diesem Modul. Beim ADC Modul muss zusätzlich noch die Schwellspannung angegeben werden, die über High oder Low entscheidet.

Das Einstellen von Registerwerten erfolgt über die Konfiguration der Port Hardware Schicht (Layer). Auf dieser Ebene kommen Modul-übergreifende Anforderungen zusammen. Beispielsweise existieren für die Parallel Schnittstelle Anforderungen von verschiedenen Modulen bzw. Signalklassen (GPIO, GPTA, SSC, CAN, etc.). Aus diesem Grund können Hardwareeinstellungen nicht Modul-spezifisch in einer Signalklasse erfolgen (siehe auch Konfigurationsprozess).

Die DIO-Konfiguration findet somit auf unterschiedlichen Konfigurationsebenen statt:

- Device Encapsulation DE: Anforderung eines Signals
- Hardware Encapsulation/HAL=SZS: Routing eines Signals
- Hardware Configuration CO: Konfiguration der Hardware Schicht

Die drei Schichten der Konfiguration **Schicht /Datei /Inhalt**

DE / *paketname.xml* / Signalname, Richtung (IN,OUT), Applizierbarkeit bzw. Einstellung der Zugriffsart (Makro oder Funktion)

HAL(Routing) / *project.xml* / Signalname, Hardware Modul, Port, Pin, [SCHWELLWERT(ADC)]

Hardware Configuration / *hardware.xml* / Hardwareeigenschaften, Registereinstellungen

Die Aufteilung der Konfiguration in verschiedene Ebenen beschränkt sich nicht nur auf das Konfigurieren digitaler Ein-

und Ausgangssignale bzw. des DIO-Moduls, auch in anderen Modulen (ADC, PWM) erfolgt die Konfiguration entsprechend obiger Klassifizierung einschließlich etwaiger Abweichungen.

- 5 Die Namen der XML-Konfigurationsdateien sind im Grunde genommen frei wählbar. Innerhalb der DE wird sicherlich eine Vielzahl an paketorientierten Konfigurationsdateien entstehen. Auf HAL-Ebene kann das Signal-Routing entweder in einer zentralen XML-Datei oder aber auf verschiedene Konfigurationsdateien verteilt werden. Hierbei muss es sich noch herausstellen, welche Aufteilung sich am besten für die Strukturierung der Projektspezifischen Konfiguration eignet. Dasselbe gilt auch für die Einstellung der Hardware bzw. Registerkonfigurationen.

15

Anforderung eines Signals

XML-Beschreibung für DIO (Paketschicht - DE)

```

20 <CONF>
    <DIO>
        <DIO_SIGNAL_REQUEST>
            <DESC>Break signal</DESC>
            <DIO_SIGNAL_NAME>E_A_BRK</DIO_SIGNAL_NAME>
25     <DIO_DIR>IN</DIO_DIR>
            <DIO_CALIBRATION>YES</DIO_CALIBRATION>
            <DIO_INIT>HIGH</DIO_INIT>
        </DIO_SIGNAL_REQUEST>
    </DIO>
</CONF>

```

Auf dieser Ebene wird vom Paket aus der DE ein Signal angefordert.

- 35 Jedes Paket kann eine oder mehrere XML-Datei(en) mit beliebigem Dateinamen anlegen und digitale Ein- und Ausgangssignale vom Core/Hwe/Dio-Modul anfordern bzw. reservieren. Dazu muss die Paketsyntax aus der XML-Beschreibung für DIO (Paketschicht - DE) in diese XML-Datei(en) kopiert und angepasst werden.

40

In obigem Beispiel wird festgelegt, dass ein Signal mit dem Namen "E_A_BRK" im Paket "DIO" verwendet wird, es eine bestimmte

Richtung hat (hier **"IN"** für Eingang) und applizierbar (Applizierbar bedeutet, dass das Signal-Routing zur Laufzeit anhand einer Signaltabelle stattfindet und somit während des Betriebs verändert werden kann. Werden die Signaldaten ausschließlich zur Compilezeit festgelegt, so können Signale zur Laufzeit nicht mehr umgeleitet werden.) ist oder nicht (hier **"YES"** für applizierbar). Die Konfiguration verschiedener Signale kann in einer XML Datei beliebig oft wiederholt werden.

10 **Anmerkung:**

Die Tags **DIO_CALIBRATION** und **DIO_INIT** sind optionale Tags. Diese sollten nur dann definiert werden, wenn aus der DE-Schicht schon im Vorfeld Anforderungen an Zugriffsart bzw. Initialisierung existieren. Bei diesen Tags handelt es sich um einen Wunsch an die Konfiguration. Werden auf der HW-Ebene abweichende Einstellungen vorgenommen, so werden die Anforderungen aus der DE-Schicht überschrieben und in der Report-Datei mitprotokolliert (siehe auch Auszug aus der Datei tmp/include_tmp/dio_report.txt).

Routing eines Signals

XML-Beschreibung für DIO (Projektschicht . HAL)

```
<CONF>
  <DIO>
    <DIO_SIGNAL_ROUTING>
      <DESC>Break signal</DESC>
      <DIO_SOURCE>E_A_BRK</DIO_SOURCE >
      <DIO_TARGET>GPIO_P8_P0_IN</DIO_TARGET>
    </DIO_SIGNAL_ROUTING>
  </DIO>
</CONF>
```

Auf dieser Ebene wird vom Projekt das Signal-Routing durchgeführt. Der Software Signalname wird sozusagen einem Hardwarepinnamen zugewiesen. Dieser Hardwarepinname ergibt sich aus der gewünschten Hardware-Ressource, welche in der Hardware Schicht konfiguriert wird (siehe Konfiguration der Hardware Schicht).

In Beispiel aus der XML-Beschreibung für DIO (Projektschicht . HAL) wird festgelegt, dass ein Signal mit dem Namen "E_A_BRK" im Paket "DIO" mit der Beschreibung (DESC) "Break signal" verwendet wird und es einem bestimmte Hardwarepinnamen "GPIO_P8_P0_IN" (Port "8", Pin "0", Direction "IN") zugeordnet werden soll ("GPIO" steht für Zugriffe auf die Parallele Schnittstelle). Die Konfiguration verschiedener Signale kann in einer XML Datei beliebig oft wiederholt werden.

10 Konfiguration der Hardware Schicht

XML-Beschreibung für DIO (Hardware Schicht)

```

15 <CONF>
    <DIO>
        <DIO_SIGNAL_IMPLEMENT>
            <DESC>hardware resource for break
            signal</DESC>
20         <DIO_MODULE>GPIO</DIO_MODULE>
            <DIO_PORT>8</DIO_PORT>
            <DIO_PIN>0</DIO_PIN>
            <DIO_DIR>IN</DIO_DIR>
            <DIO_CALIBRATION>YES</DIO_CALIBRATION>
            <DIO_INIT>HIGH</DIO_INIT>
25         </DIO_SIGNAL_IMPLEMENT>
        </DIO>
    </CONF>

```

Die Hardware-Eigenschaften des TC1775/TC1796 werden von den funktionalen Einstellungen getrennt und in einer separaten Konfigurationsschicht (HW Layer) definiert.

Jedes Projekt kann eine oder mehrere XML-Datei(en) mit beliebigem Dateinamen anlegen und Module sowie deren Ports und Pins (Endstufen und Pins) bestimmte Eigenschaften zuordnen.

35 Diese Module müssen für das Einlesen und Ausgeben digitaler Signale funktional in der Lage sein. Dazu muss die Syntax aus der XML-Beschreibung für DIO (Hardware Schicht) in diese XML-Datei(en) kopiert und angepasst werden.

40 In obigem Beispiel wird festgelegt, dass ein Modul mit dem Namen "GPIO" im Paket "DIO" mit der Beschreibung (DESC) "GPIO_P8_P0_IN" verwendet wird ("GPIO" steht für Parallele

Schnittstelle). Von diesem Modul wird die Ressource (Port "8", Pin "0") angefordert und der Applikationsidentifizier "YES" sowie der Initialisierungswert "HIGH" zugewiesen (sofern dieses an dieser Stelle für einen Eingang Sinn macht). Die Konfiguration verschiedener Signale kann in einer XML Datei beliebig oft wiederholt werden.

Generischer Hardwarepinnamen

Der Hardwarepinname ist ein rein generischer Name, welcher einer Hardwareressource eindeutig zugeordnet und nach dem folgenden Muster zusammengestellt wird.

`<Modulname>_P<Portnummer>_P<Pinnnummer>_<Richtung>`,

z.B. `GPIO_P8_P0_IN`.

Der Hardwarepinname lässt sich nach einem Konfigurationslauf der DIO Report Datei entnehmen (siehe nachfolgend Generischer Hardwarepinnamen).

Auszug aus der Datei tmp/include_tmp/dio_report.txt

DIO Report
Signal Map - Digital Input/Output Signals

Signal name: OUTPUT
HW Signal: GPIO_P9_P0_OUT
Module: GPIO
Port: 9
Pin: 0
Calibration: YES

Signal name: INPUT
HW Signal: GPIO_P9_P4_IN
Module: GPIO
Port: 9
Pin: 4
Calibration: YES

Signal name: INPUT2
HW Signal: GPIO_P9_P8_IN
Module: GPIO
Port: 9
Pin: 8

Calibration: YES

Anmerkungen

Es wird deutlich, dass die XML-Elemente `<DIO_CALIBRATION>` und `<DIO_INIT>` in jeweils zwei unterschiedlichen Schichten vorkommen. Beide XML-Elemente sind optional, d.h. falls diese nicht explizit angegeben werden wird von folgenden Default-Einstellungen ausgegangen:

```
10 <DIO_CALIBRATION>NO</DIO_CALIBRATION>
    <DIO_INIT>LOW</DIO_INIT>
```

Kommen diese XML-Elemente sowohl in der DE-Schicht (Anforderung eines Signals), als auch bei der Konfiguration der Hardware Schicht vor, so sind die Einstellungen in der Hardwareschicht auf Projektebene höherprior. Hierbei wird angenommen, dass die Projektebene spezifische Einstellungen vornehmen und auch gegenüber der DE-Schicht durchsetzen kann. Etwaige Überlappungen werden mitprotokolliert und können nach dem Konfigurationslauf eingesehen werden.

Registrieren der XML-Konfigurationsdatei

Nach der Durchführung der Konfiguration gemäß Anforderung eines Signals und Routing eines Signals müssen die gespeicherten XML-Dateien in die jeweiligen Makefiles unter dem Element CONF eingetragen werden, siehe Abbildung 5: Makefile mit eingetragener XML-Konfigurationsdatei.

```
30 Makefile mit eingetragener XML-Konfigurationsdatei
    #-----
    # Module definition
    #-----
    NAME                := dio
35  INTERFACE            := dio.h
    CONF                := dio_request.xml dio_routing.xml dio_implement.xml
    CSOURCES              := dio.c gpio.c
    ASMSOURCES            :=
    IMPLEMENTATION        :=
40  DATA                :=
    SUBDIRS              := diocfg
```

Eingabe der Konfigurationsdaten

Die Eingabe der Konfigurationsdaten erfolgt über einen XML-Browser (z.B. XMetal). Anhand einer *Document Type Definition* (DTD) für die Konfiguration des Miniprojektes wird die XML-Struktur vorgegeben. Die DTD für die Konfiguration des Miniprojektes liegt auf dem edc_hwe vob im Miniprojekt unter `\scripts\xml\medc17_conf.dtd`. Diese DTD wird von den Entwicklern der Core Pakete (Module) entsprechend den bereit gestellten, konfigurierbaren Daten sukzessive erweitert. Die XML-Konfigurationsdateien müssen über diese DTD plausibilisiert werden.

15 Konfigurationsprozess

Durch die Aufteilung in verschiedene Konfigurationsebenen ist es möglich, ohne detaillierte Kenntnisse über die Konfigurationsweise anderer Schichten, auf unterschiedlichen Ebenen Einstellungen vorzunehmen.

20 Dabei müssen nur die Konfigurationsparameter angegeben werden, die für die zu konfigurierende Schicht auch tatsächlich relevant sind.

Das entsprechende Schichtenmodell ist in Figur 11 dargestellt.

25 Anhand der Beschreibung digitaler Ein- und Ausgangssignale lässt sich dieser Prozess recht anschaulich erläutern. In der obersten Schicht (DE) wird eine Signalanforderung an die Signalklasse DIO gestellt. Neben der Richtung, dem Initialisierungswert und dem Parameter Applizierbarkeit, ist es der Signalname, über den die Anforderung aus der DE mit dem Routing in der HAL verknüpft wird. In der HAL kann nun der Signalname auf die entsprechende Ressource geroutet werden. Das Mapping mit den Einstellungen in der HWL erfolgt über den Signalnamen bzw. Hardwarepinnamen des Pins. In der Hardware Layer muss neben dem Modulnamen auch der entsprechende Port bzw. Pin angegeben werden.

35 Die Eigenschaften für die PORT Hardware z.B. Registereinstellungen werden separat über die PORT-

Konfiguration vorgenommen. Die Konfiguration der einzelnen Schichten wird in der Regel auf verschiedene XML-Dateien verteilt. Die Konfigurationsgeneratoren haben die Aufgabe die konfigurierten Daten zu überprüfen und diese zu
5 plausibilisieren.

Konfigurationsgeneratoren

Nach dem Aufruf des "make" bzw. "make xml2conf" -Kommandos im Miniprojekt wird anhand der registrierten XML-

10 Konfigurationsdateien eine Dateiliste für die angemeldeten Konfigurationsdateien generiert. Darauf basierend sammelt der zentrale XML-Parser für die Konfiguration (ehemals core_parse.pl) alle Konfigurationsdateien bzw. .Daten ein. Im Anschluss an den Parsing-Prozess werden die individuellen
15 Konfigurationsgeneratoren gestartet und die entsprechenden *.c- und *.h-Konfigurationsdateien erzeugt.

Die Konfigurationsgeneratoren prüfen zusätzlich die Datenkonsistenz und plausibilisieren die Benutzerdaten. Nach erfolgreichem Beenden des "Makelaufs" können die konfigurierten
20 Signale im Steuergeräte-Code verwendet werden.

21.02.2003

5 Robert Bosch GmbH, 70442 Stuttgart

Ansprüche

10

1. Steuergerät (100) mit wenigstens einem Prozessor (100a-1) und wenigstens einem Speicherelement (100a-2) zum Steuern eines Antriebsaggregates (300), insbesondere einer Brennkraftmaschine, eines Fahrzeugs mit Hilfe einer
15 zwischen das Steuergerät (100) und dem Antriebsaggregat (300) geschalteten Sensor-/Aktor-Konfiguration (200), wobei die Steuerung durch eine Kommunikation zwischen einer Vielzahl von in dem Speicherelement (100a-2) hinterlegten Funktionseinheiten erfolgt;

20

gekennzeichnet durch

ein zweites hardwarenäheres Modul (CO), das mit einem
25 dritten hardwareferneren Modul (DE) über eine Signalzuordnungsschicht (SZS=HAL) verbunden ist, welche die digitalen Signale des einen Moduls dem anderen zuordnet, wobei hardwarenäher und hardwareferner auf den Prozessor bezogen ist.

30

2. Steuergerät nach Anspruch 1, dadurch gekennzeichnet, dass ein erstes Modul (ASW), in dem diejenigen Funktionseinheiten zusammengefasst sind, die zur Beeinflussung des Antriebsaggregates im Ansprechen auf einen Benutzerwunsch auf physikalischer Ebene dienen,

enthalten ist.

3. Steuergerät nach Anspruch 1, dadurch gekennzeichnet, dass das zweite Modul (CO), derart ausgebildet ist, dass in dem diejenigen Funktionseinheiten zusammengefasst sind, welche eine individuelle Programmierung der Hardware des Steuergerätes in der Weise ermöglichen, dass die Hardware in die Lage versetzt wird, mit den Modulen des Steuergeräts (100) zu kommunizieren und welche die Abarbeitung von Funktionen der Funktionseinheiten in den Modulen zeitlich koordinieren; und
- das dritte Modul (DE), derart ausgebildet ist, dass in dem diejenigen Funktionseinheiten zusammengefasst sind, die eine individuelle Anpassung der verwendeten Sensor-/Aktor-Konfiguration (300) an das Steuergerät (100) in der Weise ermöglichen, dass zwischen den einzelnen Sensoren oder Aktoren der Konfiguration eine Kommunikation mit den übrigen Modulen des Steuergerätes möglich ist; und
- wobei zwischen den Modulen (CO, DE) Modul-Schnittstellen (M1 ... M5) für eine modul-übergreifende Kommunikation vorgesehen sind.
4. Steuergerät (100) nach Anspruch 2 und 3, dadurch gekennzeichnet, dass das erste Modul (ASW) aufweist:
- eine Fahrzeug-Komponente (VF), in welcher diejenigen Funktionseinheiten zusammengefasst sind, die nicht spezifisch für einen bestimmten verwendeten Typ von Antriebsaggregat (300) sind; und
 - eine Antriebsaggregate-Komponente (EF), in welcher diejenigen Funktionseinheiten zusammengefasst sind, die für den verwendeten Typ von Antriebsaggregat (300) spezifisch sind.

5. Steuergerät (100) nach einem der vorangegangenen Ansprüche, dadurch gekennzeichnet, dass das zweite Modul (CO) aufweist:

5 - eine Infrastruktur-Komponente (IS), in welcher diejenigen Funktionseinheiten zusammengefasst sind, welche grundlegende Dienste anbieten oder repräsentieren, auf die andere Funktionseinheiten zugreifen können; und

10 - eine Hardwarekapsel-Komponente (HWE), in welcher diejenigen Funktionseinheiten zusammengefasst sind, welche eine individuelle Programmierung der Hardware (100a) des Steuergerätes (100) in der Weise ermöglichen, dass die Hardware in die Lage versetzt wird, mit den Modulen (ASW, CO, DE, CD) des Steuergeräts (100) zu kommunizieren.

15 6. Steuergerät (100) nach Anspruch 5, dadurch gekennzeichnet, dass die Infrastruktur-Komponente (IS) vorzugsweise die Funktionseinheiten Dienstbibliotheken (IS-1), Ablaufsteuerung (IS-2), Diagnosemanager (IS-3) und/oder Überwachungskonzept (IS-4) umfasst.

20 7. Steuergerät (100) nach einem der vorangegangenen Ansprüche, dadurch gekennzeichnet, dass das Steuergerät (100) ein viertes Modul (CD) aufweist, in dem diejenigen Funktionseinheiten zusammengefasst sind, die eine direkte Ansteuerung von speziellen Sensor-Aktor-Konfigurationen mit
25 komplexen Schnittstellen zu dem Steuergerät durch das erste Modul ermöglichen.

8. Steuergerät (100) nach einem der vorangegangenen Ansprüche, dadurch gekennzeichnet, dass die Funktionseinheiten, die Komponenten und/oder die Module
30 sowie die Schnittstellen zwischen ihnen zumindest teilweise als Computerprogramm ausgebildet sind.

9. Computerprogramm für ein Steuergerät (100) gemäß einem der Ansprüche 1 - 7 zum Steuern eines Antriebsaggregates (300) eines Fahrzeugs, umfassend einen Programmcode, der geeignet ist, die Funktionseinheiten, Komponenten oder
- 5 Module abzubilden und eine Kommunikation zwischen diesen zum Zwecke der Steuerung des Antriebsaggregates zu realisieren.

5

21.02.2003

Robert Bosch GmbH, 70442 Stuttgart

10 Steuergerät und Computerprogramm zum Steuern eines
Antriebsaggregates eines Fahrzeugs

Zusammenfassung

15

Die Erfindung betrifft ein Steuergerät (100) und ein
Computerprogramm zum Steuern eines Antriebsaggregates (300)
eines Fahrzeugs. Traditionell sind in einem derartigen
Steuergerät (100) zahlreiche Funktionseinheiten vorgesehen,
20 zum Beispiel eine Funktionseinheit Antrieb,
Motorkoordinator, Diagnosemanager et cetera. Um
insbesondere im Falle eines Wechsels des von dem
Steuergerät anzusteuernenden Antriebsaggregates nicht alle
oder einen Großteil der Funktionseinheiten, sondern nur die
für das neue Antriebsaggregat relevanten Funktionseinheiten
austauschen zu müssen, wird erfindungsgemäß eine
Modularisierung dieser Funktionseinheiten vorgeschlagen.
Diese Modularisierung sieht insbesondere vor, dass
diejenigen Funktionseinheiten zusammengefasst werden, die
30 eine individuelle Programmierung der Hardware des
Steuergerätes in der Weise ermöglichen, dass die Hardware
in die Lage versetzt wird, mit den Modulen des
Steuergerätes 100 zu kommunizieren. Figur 1

Fig. 1

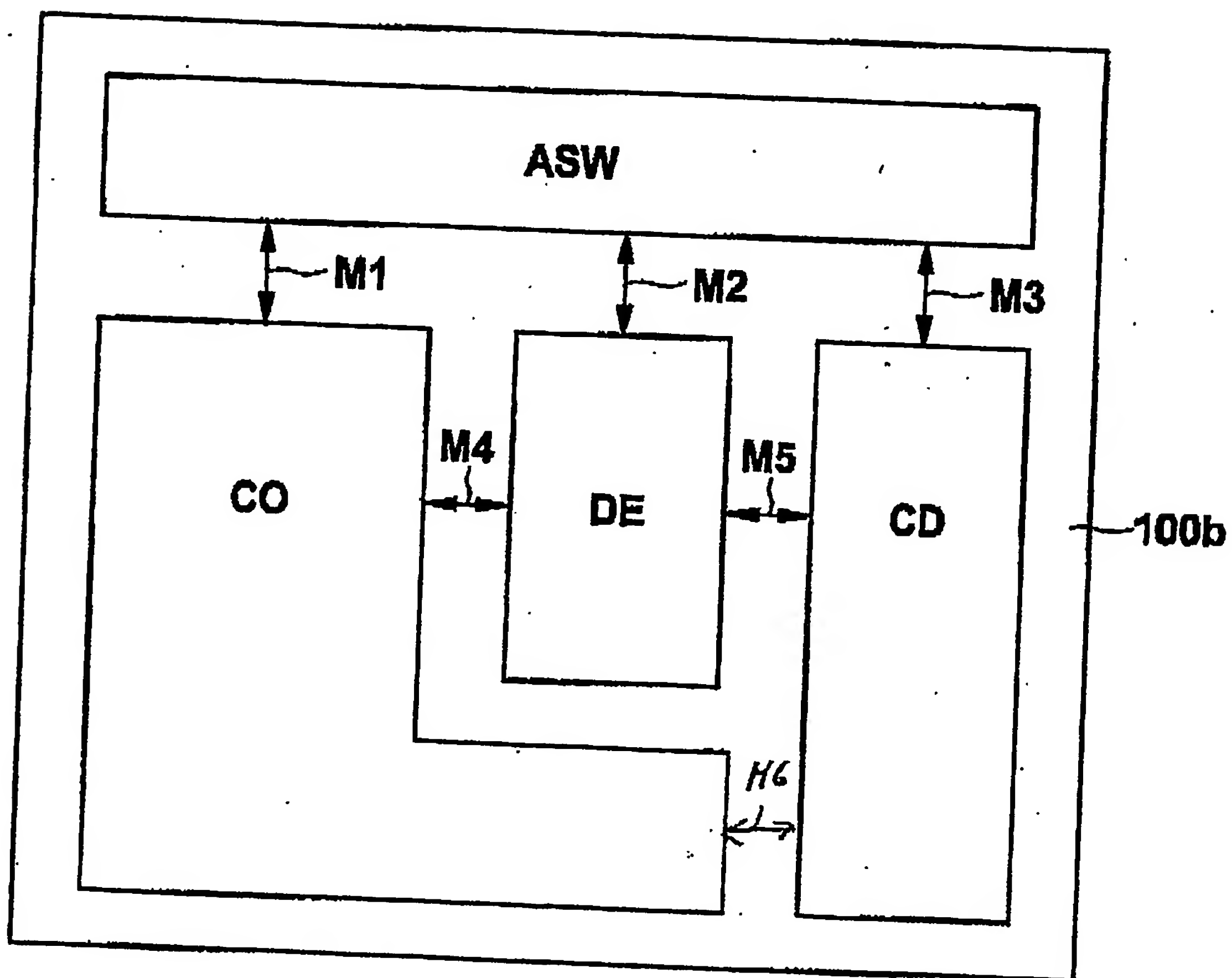


Fig. 2

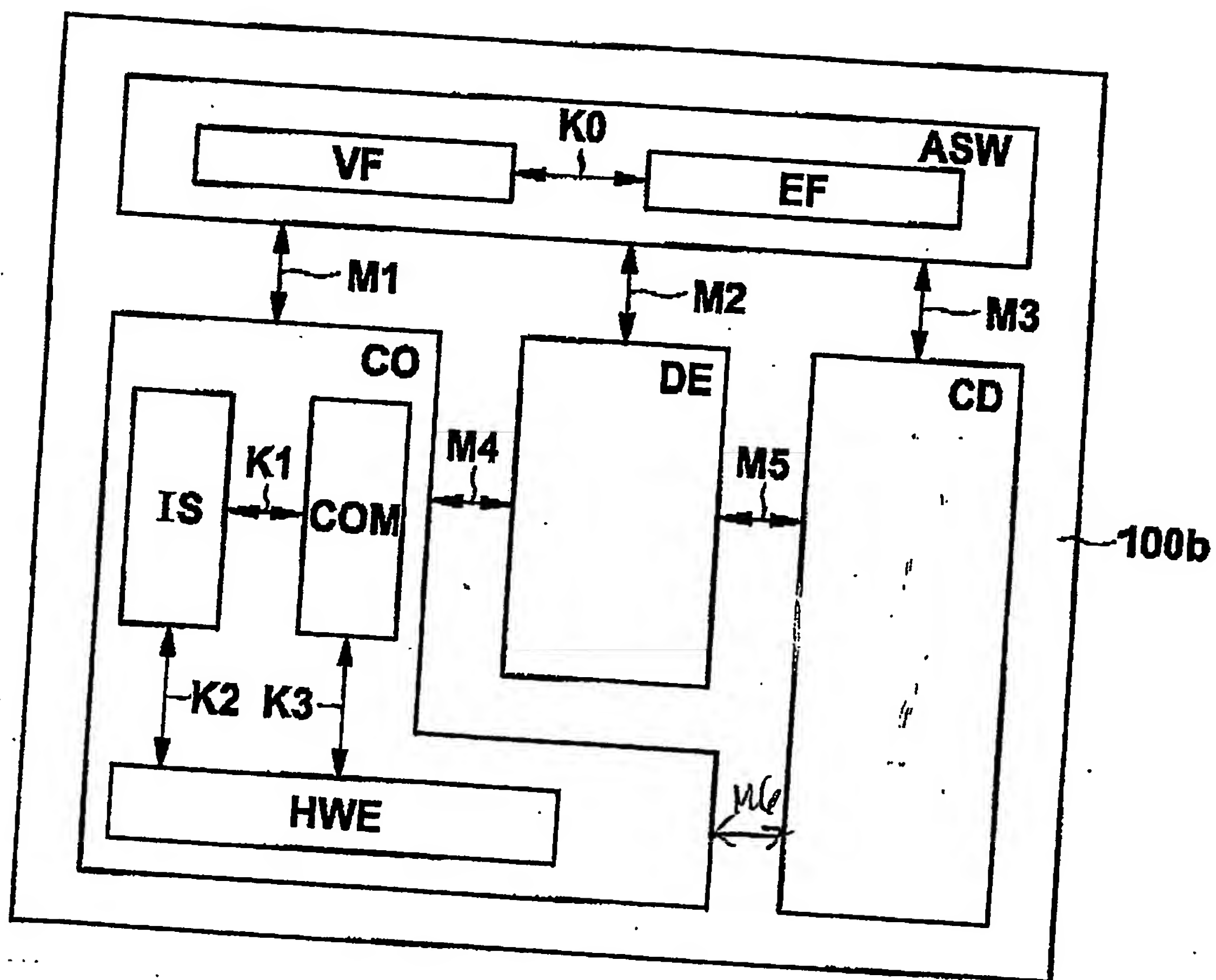


Fig. 3

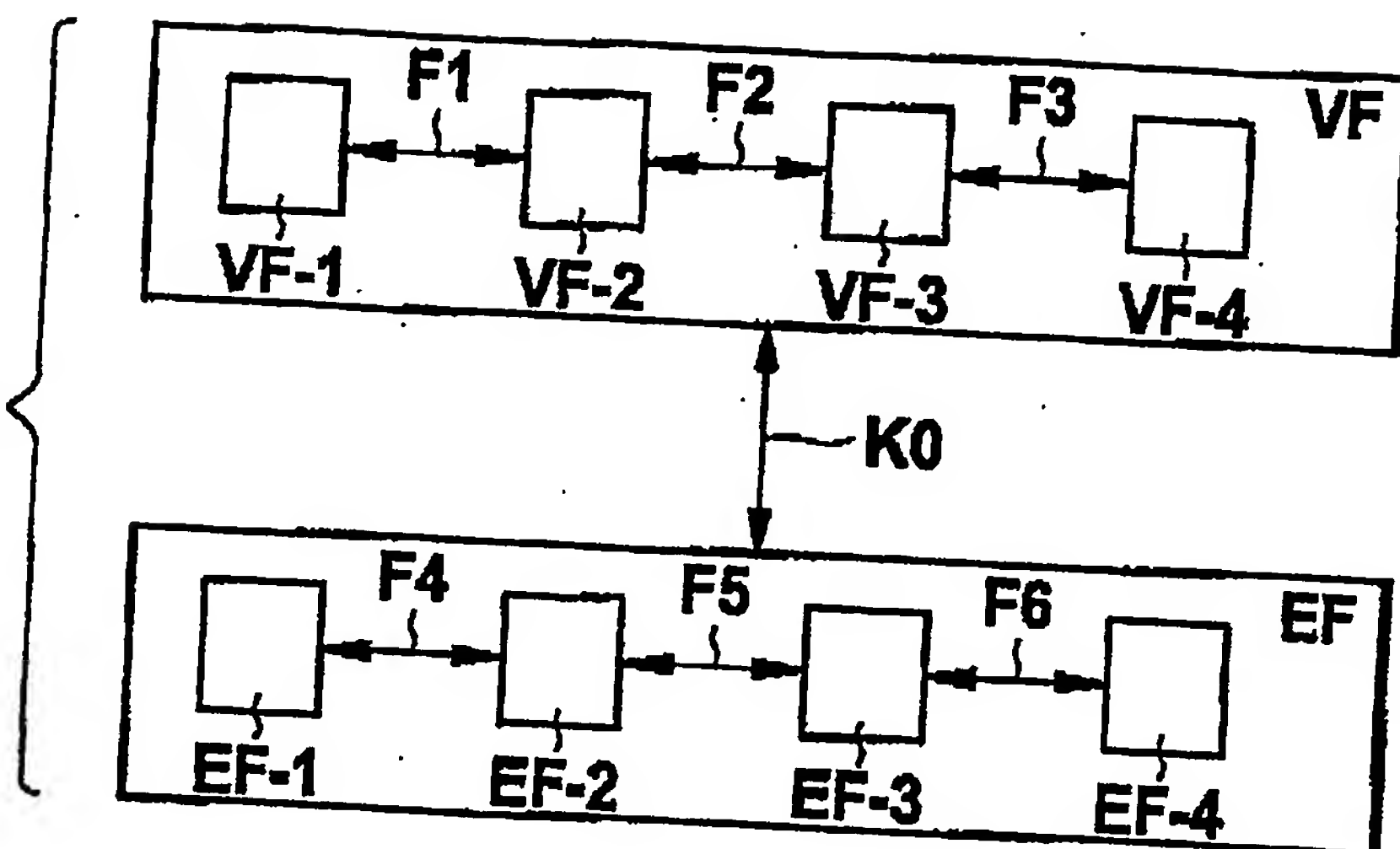


Fig. 4

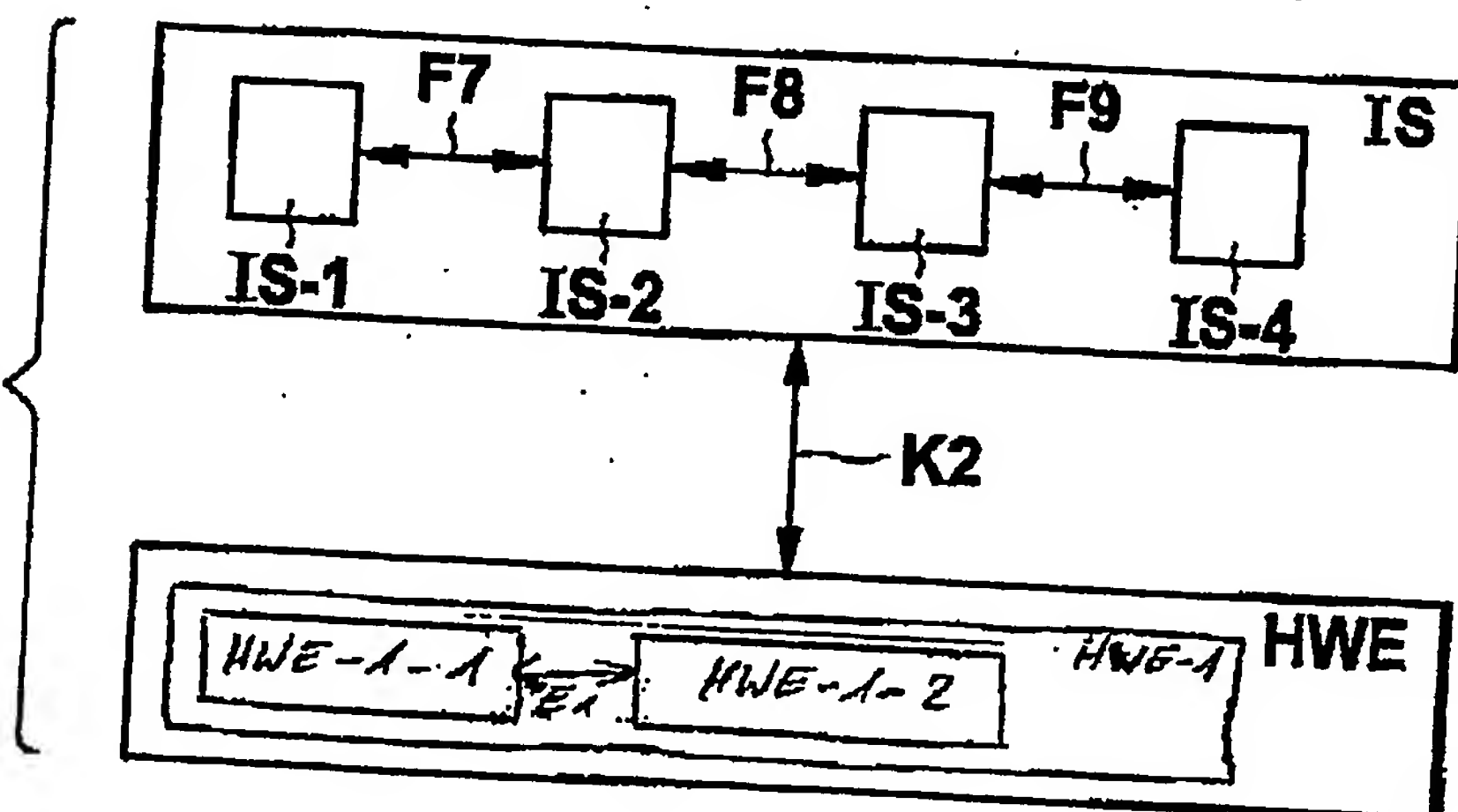
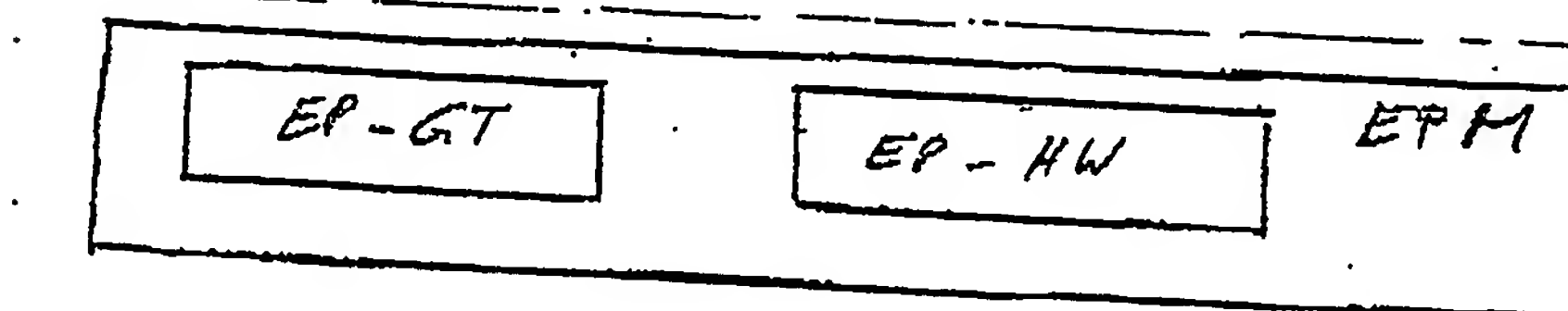


Fig. 5



The diagram illustrates the architecture of the CO system, enclosed within a dashed boundary labeled 'CO' in the bottom-left corner. The system consists of several interconnected components:

- IS DSM** (Intelligent System Decision Support Module): Located in the upper-left area of the CO boundary.
- COM** (Command Module): Located in the lower-left area of the CO boundary.
- DE** (Decision Engine): Positioned in the center of the CO boundary.
- ASW** (Air Support Weapon): Located outside the CO boundary at the top.
- HWE** (Hostile Weapon Effect): Located outside the CO boundary at the bottom.

The data flow and interactions are as follows:

- A curved arrow points from **IS DSM** to **DE**.
- A curved arrow points from **COM** to **DE**.
- A curved arrow points from **DE** back to **COM**.
- A curved arrow points from **DE** to **ASW**.
- A curved arrow points from **ASW** back to **DE**.
- A curved arrow points from **DE** to **HWE**.
- A curved arrow points from **HWE** back to **DE**.

Two specific interaction loops are highlighted with labels:

- M2**: A curved arrow pointing from **ASW** to **DE**.
- M4**: A curved arrow pointing from **DE** to **COM**.

The diagram illustrates a relationship between two systems. On the left, a dashed box labeled 'IS' contains a solid rectangle labeled 'IS-3'. On the right, a dashed box labeled 'HWE' contains a solid rectangle labeled 'HWE-1'. A curved arrow points from 'HWE-1' to 'IS-3', indicating a flow or dependency from the HWE system to the IS system.

Fig. 8

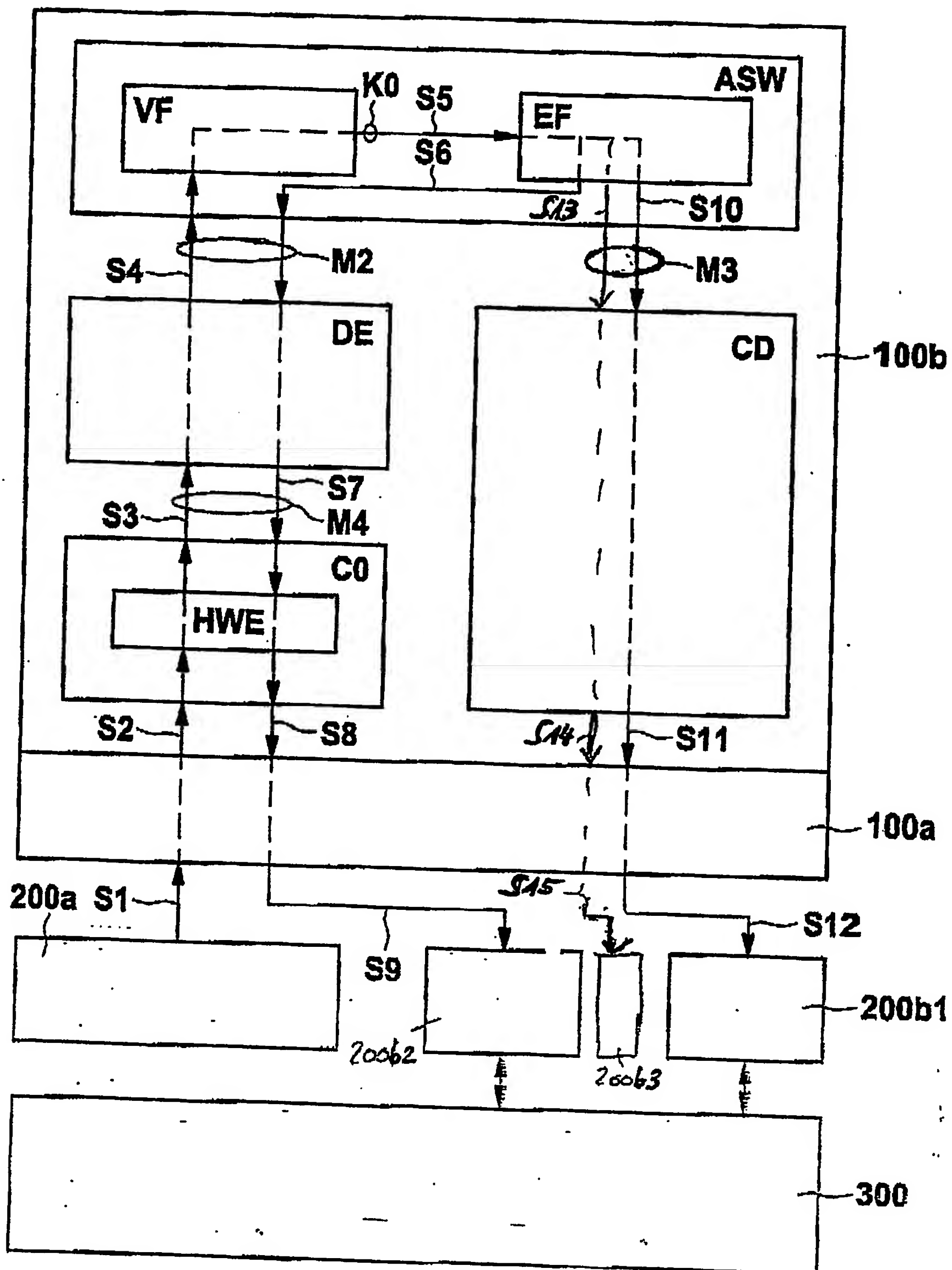
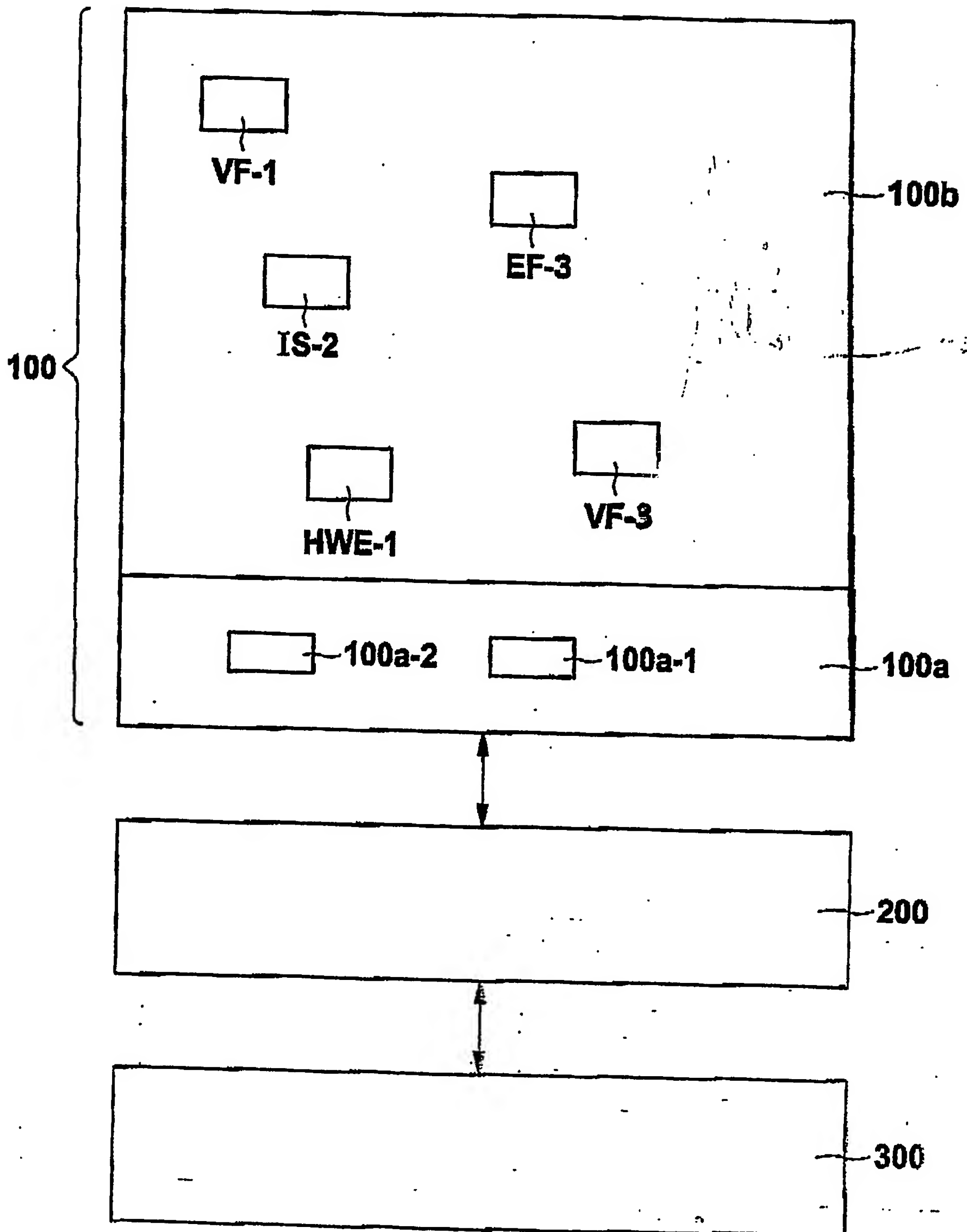


Fig. 9



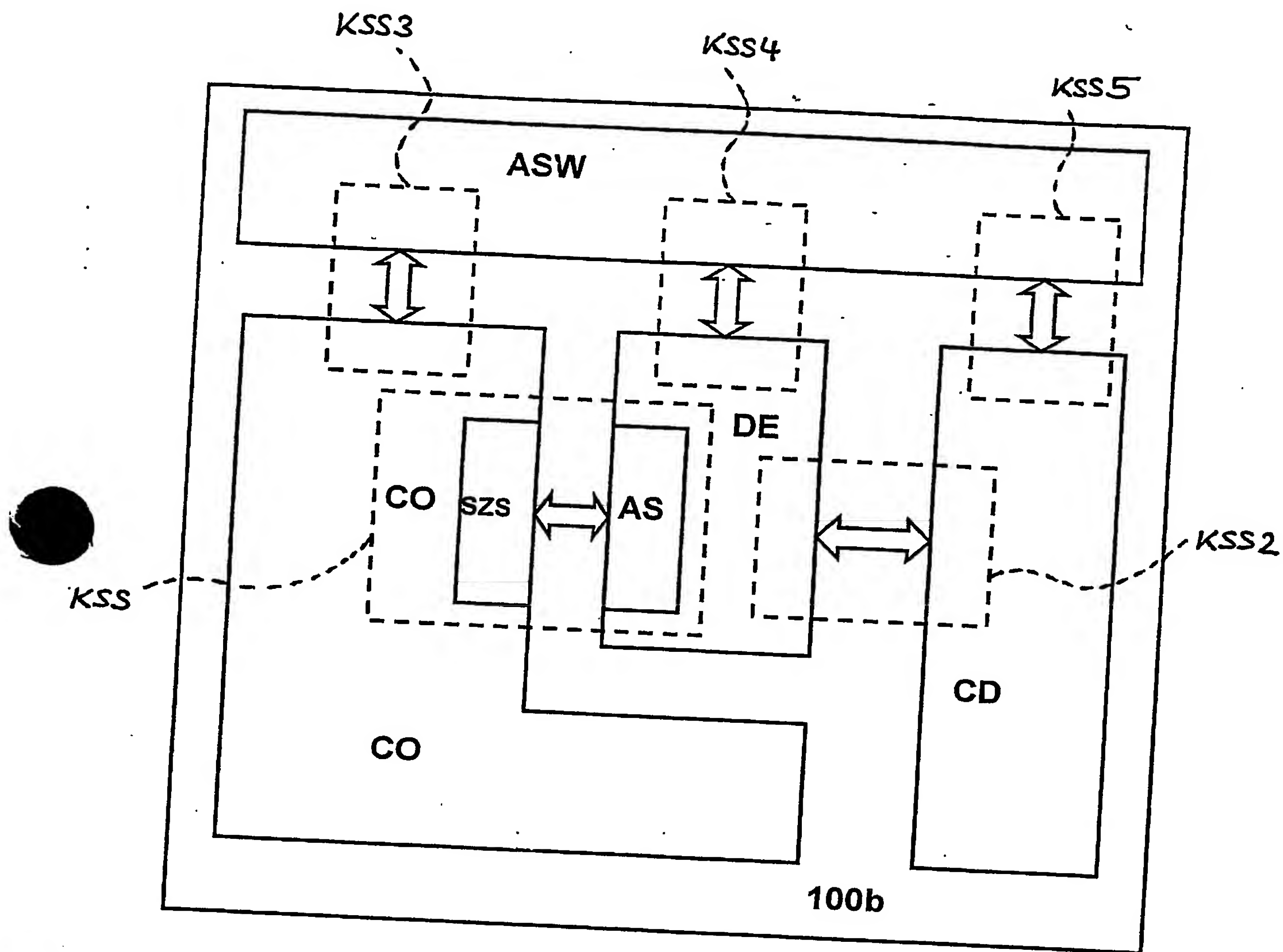


Fig. 10

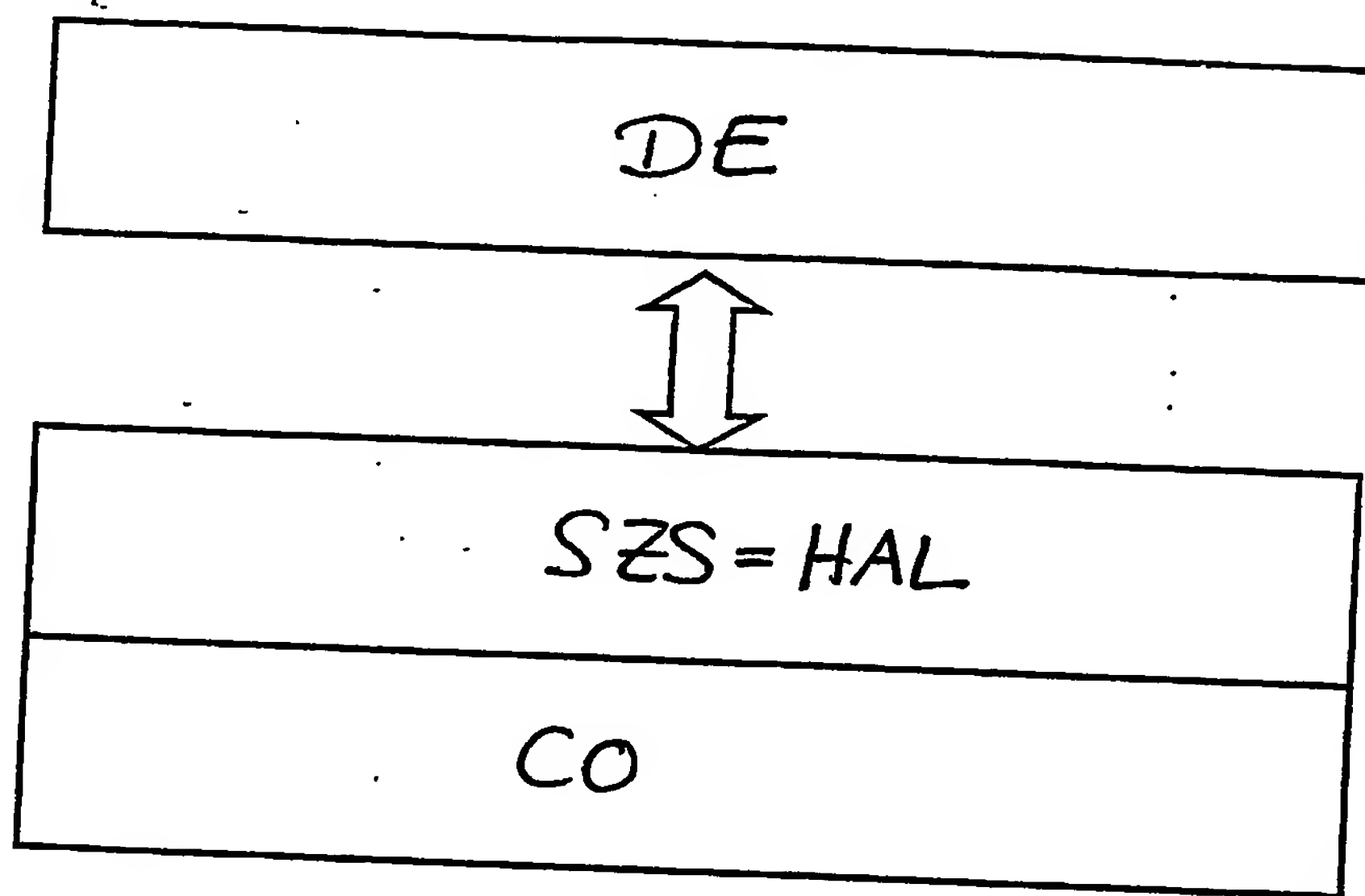


Fig 11